

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Remote Vehicle System Management

Inventor(s):

Eugene Luskin

Andrew Petrochuk

David Kizhnerman

ATTORNEY'S DOCKET NO. MS1-1726US

CLIENT'S DOCKET NO. 305545.01

REMOTE VEHICLE SYSTEM MANAGEMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is related to concurrently filed U.S. Patent Application No. _____ entitled "SMART VEHICLE VIDEO MANAGEMENT", and U.S. Patent Application No. _____ entitled "ENHANCED VEHICLE EVENT INFORMATION", both of which are assigned to the Assignee of the present application.

TECHNICAL FIELD

The described subject matter relates to vehicle systems. More particularly, the subject matter relates to remote vehicle system management.

BACKGROUND

As automobiles and other vehicles advance, more information is used and generated by the vehicles both in terms of quantity and types of information. For example, modern automobiles store a user profile that can include data about a driver's preferred seat position, preferred temperature, and preferred radio stations. Infotainment systems, global positioning systems and on-board diagnostic systems in automobiles also provide and use large amounts of data. Such vehicle systems can greatly improve the transportation experience offered by the vehicle. Typically, each type of vehicle data is handled independently by an independent vehicle system related to the particular type of data.

To retrieve data from a vehicle or update data in a vehicle system, a physical connection is typically made to the vehicle system, whereby the data can

1 be accessed. For vehicle system data to be updated or accessed, a user must be
2 physically present at the vehicle or system. For example, to retrieve data from an
3 on-board diagnostic (OBD) system, a mechanic connects an OBD reader to the
4 OBD system to read the OBD codes. As another example, user media such as
5 music can be loaded using compact disk (CD) player installed in the vehicle. As
6 another example, user profile data can be changed by a driver through
7 manipulation of certain buttons or use of a user interface. Each system typically
8 has a dedicated, separate interface. Certain vehicle systems usually have no
9 information about the presence of each other; i.e., these vehicle systems are
10 typically not aware of each other. For example, GPS is not aware of a vehicle-
11 based infotainment system, and an OBD system is not aware of GPS system, etc.

12 Another drawback to current vehicle systems is that user interfaces for
13 vehicle systems are typically very limited, partly because the separate vehicle
14 systems do not interact. Various types of vehicle system data are available in
15 modern vehicles, but the vehicle systems operating on the vehicle system data have
16 been developed and deployed separately. Thus, the information from various
17 systems is not presentable or updateable remotely or through one common user
18 interface.

19 20 SUMMARY

21 Implementations of systems and methods described and claimed herein
22 solve the discussed problems, and other problems, by providing for remote vehicle
23 system management. A server in a vehicle accesses data related to separate vehicle
24 systems. The server can generate a network document, which is accessible by a
25

1 remote client. The server can respond to network requests to access or update
2 vehicle system data.

3 An implementation of a method includes generating a browsable network
4 document by a vehicle-based server, the browsable network document having
5 vehicle system data. The method may further include receiving a network request
6 for vehicle system data. The method may further include receiving vehicle system
7 configuration data related to a configurable vehicle system and updating the
8 configurable vehicle system.

9 An implementation of a vehicle includes a vehicle-based web server
10 generating a web page including vehicle system data. A network communications
11 module transmits the network document over a network. The web server may
12 further receive a network request for vehicle system data and responsively generate
13 a web page having the requested vehicle system data. The web server may also
14 receive configuration data related to a configurable vehicle system and update the
15 configurable vehicle system. The web server may also encrypt network documents
16 and decrypt network requests to protect sensitive vehicle information and change
17 requests.

18 An implementation of a vehicle management system includes at least one
19 interface to a vehicle system and a network server receiving data from the vehicle
20 system. The network server generates a browsable network document having the
21 vehicle system data. The network server may generate the network document in
22 response to a client request for the vehicle system data. The at least one interface
23 may include an interface to at least one of a GPS, an On-Board Diagnostics (OBD)
24 system and a media player.
25

1 An implementation of a computer program product provides a computer
2 program storage medium readable by a computer system and encoding a computer
3 program that accesses vehicle system data for configuring a system in a vehicle or
4 communicating data from a system in the vehicle to a client in a browsable
5 network document having vehicle system data.

6 7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 Fig. 1 illustrates an exemplary operating environment in which a remote
9 vehicle system management scheme may be employed.

10 Fig. 2 illustrates a plan view of a vehicle operable to employ remote vehicle
11 system management.

12 Fig. 3 illustrates a block diagram of an exemplary vehicle-based computer
13 system that enables remote vehicle system management.

14 Fig. 4 illustrates an exemplary arrangement of vehicle systems, vehicle
15 system data, and a relational database application that can collect and relate vehicle
16 system data.

17 Fig. 5 illustrates an exemplary screenshot of the web page that can be
18 generated by a vehicle-based server.

19 Fig. 6 illustrates a flowchart having exemplary operations for remotely
20 managing one or more vehicle computer systems.

21 Fig. 7 illustrates a flowchart having exemplary operations for remotely
22 configuring data for one or more configurable vehicle computer systems.

23 Fig. 8 illustrates a suitable computing environment in which vehicle
24 systems management may be implemented.

DETAILED DESCRIPTION

Overview

Exemplary implementations of methods, systems, devices, computer program products, and data structures are disclosed for remotely managing one or more systems in a vehicle. Traditional systems and methods for managing vehicle systems involve being physically present at the vehicle and creating a physical connection to manage systems through user interfaces that are different for each vehicle system. Implementations described herein provide for remote management of vehicle-based systems through a vehicle-based server that can generate a network document and receive network requests for data related to one or more vehicle-based systems.

Exemplary Operating Environment

Fig. 1 illustrates an exemplary operating environment 100 in which a remote vehicle system management scheme may be employed. The environment 100 includes a vehicle 102 that includes one or more vehicle systems. As used herein a vehicle system is any on-board system that provides data about operation of the vehicle. Examples of vehicle systems are control systems, diagnostics systems, entertainment systems, and navigation systems.

A vehicle-based computer (not shown) located within or on the vehicle 102 can communicate data related to the vehicle system(s) over a network 104. As illustrated, the vehicle 102 may communicate with a satellite 108 and/or a cell tower 106 to access the network 104. Via the network 104, the vehicle-based computer can communicate with remote computing devices, such as, but not

1 limited to, a remote client 110 (e.g., a desktop computer) or a remote server
2 computer 112.

3 The network 104 may include a number of interconnected sub-networks.
4 For example, the network 104 may be the Internet. The network 104 may also
5 include a satellite, telephone land-line, or wireless network. The network 104
6 facilitates communication among computing devices using a communication
7 protocol. Exemplary communication protocols are TCP/IP, HTTP, and SOAP.

8 Regardless of the particular network 104 or communication protocol used,
9 one or more computer systems in the vehicle 102 can use the network 104 to
10 communicate with the remote server 112 and the remote client 110, as long as the
11 remote server 112 and remote client 110 support the communication protocol.
12 Although illustrated as desktop computers, the remote client 110 and remote server
13 112 may be implemented with other known computing devices, such as, but not
14 limited to, handheld computers, laptops, cell phones, Personal Digital Assistants
15 (PDAs), or others. Such devices typically include a network application, such as,
16 but not limited to, INTERNET EXPLORER from MICROSOFT Corporation, that
17 enables the devices to transmit and receive data to and from the network 104.

18 A vehicle-based computer can generate a browsable network document,
19 such as a web page definition, which includes vehicle system data from vehicle
20 systems. The browsable network document can be transmitted to the remote server
21 112 or the remote client 110, where the vehicle system data may be browsed.
22 Network applications typically include a browser utility that enables a user of the
23 remote server 112 or remote client 110 to view electronic documents from the
24 network 104. Systems in the vehicle 102 can generate such browsable documents
25 for distribution over the network 104. Such browsable documents can include

1 vehicle system data, such as, but not limited to, Global Positioning System (GPS)
2 data, user configuration data, or On-Board Diagnostics II (OBD-II) data, from
3 systems in the vehicle 102.

4 The remote client 110 or server 112 may also be enabled to upload data to
5 the vehicle-based computer in the vehicle 102. Data that is uploaded to the vehicle
6 102 may be used by one or more vehicle systems in the vehicle 102. Such data
7 may include user data, updates, system configurations, or settings. For example, a
8 GPS or mapping system in the vehicle 102 may be updated with the most up-to-
9 date maps of city streets or facilities, etc. As another example, a user of the
10 vehicle 102 can upload music, videos, or other types of media to the vehicle 102.
11 Systems in the vehicle 102 receive and store the uploaded data for use in the
12 vehicle 102.

13 In addition, the systems in the vehicle 102 can receive requests from the
14 network 104 for particular information from the vehicle 102. For example, a
15 browsable web page from the vehicle 102 may include entry fields in which a user
16 of the remote client 110 can enter a request for a particular type or types of vehicle
17 data, such as GPS data or OBD data. The systems in the vehicle 102 can generate
18 a browsable network document that includes the requested vehicle data. In one
19 implementation, a vehicle computer can combine different data types from
20 different systems in the vehicle 102 to create a more informative presentation of
21 vehicle systems, than may otherwise be possible using each system separately.

22 A remote computer management scheme as described herein may be
23 beneficially implemented in any mobile vehicle. Thus, the vehicle 102 is not
24 limited to any particular type of vehicle. For example, the vehicle 102 may be an
25 automobile. As another example, the vehicle 102 may be a farm tractor. As yet

1 another example, the vehicle 102 may be a grader, a back-hoe, a paver, or other
2 heavy equipment. Other examples of vehicles include boats, airplanes, or
3 helicopters.

4 Fig. 2 is a plan view of a vehicle 200 having systems operable to employ
5 remote vehicle computer management. The vehicle 200 includes a web server
6 computer 202 that is network enabled for communicating on a network. As such,
7 the server 202 is operable to collect data from one or more vehicle systems and
8 generate browsable network documents including the collected data. In addition,
9 the web server 202 is operable to receive data from a network and store the
10 received data in memory for use by the systems in the vehicle 200.

11 Exemplary vehicle systems, an On-Board Diagnostics II (OBDII) system
12 204, a GPS 206, and a video camera 208 are installed in the vehicle 200. Such
13 systems generate and/or use associated data to facilitate tasks for a driver, other
14 occupants of the vehicle, or remote clients of the server computer 202. For
15 example, the OBDII system 204 generate error codes or event codes indicative of
16 vehicle system errors or events that can be presented to the driver of the vehicle, or
17 a mechanic who is remotely logged-in to the server computer 202.

18 As another example, the GPS 206 may employ map data that can be
19 downloaded from a network and illustrated to occupants of the vehicle 200. As a
20 further example, video images from the video camera 208 may be presented to
21 occupants of the vehicle 200 or transmitted to a remote client over a network. As
22 shown, the video camera 208 is directed to capture a rear view 210 behind the
23 vehicle 200. In other implementations, the video camera 208 may be directed
24 toward the front or sides of the vehicle 200 to capture other views. While not
25

1 shown, other systems, such as obstacle sensors or a vehicle security system, may
2 be installed in or on the vehicle 200 and communicate with the server 202.

3 A local client 212 can be installed in the vehicle 200 and used by occupants
4 of the vehicle 200. The local client 212 may be a portable computing device, such
5 as a handheld computer, a PDA, a cell phone, or a laptop. The local client 212
6 may also be mounted in or on the vehicle 200. Media devices 214 are input/output
7 devices through which a vehicle occupant can interact with the local client 212
8 and/or the web server 202. Exemplary media devices include speakers and video
9 screens. Thus, for example, a video screen can show a map of the current position
10 of the vehicle 200 from the GPS system 206.

11 The web server 202 may also utilize media devices for data input/output.
12 Like the client 212, the web server 202 may be a portable device or arranged in a
13 casing or housing that is installed in one of various locations in the vehicle 200.
14 One exemplary housing has a standardized size expressed in terms of Deutsche
15 Industry Normen (DINs). The housing may be installed in the dashboard of the
16 vehicle 200, under a floor board of the vehicle 200, in the trunk of the vehicle 200,
17 or other convenient location, from which the web server 202 may communicate
18 with vehicle systems.

19 Fig. 3 is a block diagram of an exemplary vehicle-based computer 300 that
20 enables remote vehicle system management. The vehicle-based computer 300
21 includes one or more vehicle system interfaces for interacting with the vehicle
22 systems. The vehicle-based computer 300 includes memory, such as a vehicle
23 information store 302, for storing data associated with the one or more vehicle
24 systems. A server application 304 communicates with the system interfaces to
25 update and upload vehicle system data. Using the interfaces and memories, the

1 server application 304 can retrieve and manage data generated and/or used by the
2 vehicle systems.

3 Exemplary vehicle systems include an OBDII system 306, a GPS system
4 308, and a video source 310, as shown in Fig. 3. The OBDII system 306 detects
5 OBDII vehicle events and errors and outputs codes (herein referred to as raw
6 OBDII data) representing the errors and events when they occur. The GPS system
7 308 is in communication with one or more satellites to determine the current
8 location of the vehicle and generate vehicle location data, such as latitude and
9 longitude.

10 The video system 310 includes one or more video capturing devices, such
11 as video cameras, which generate images of views around the vehicle. Many other
12 vehicle systems in addition to those shown in Fig. 3 may communicate with the
13 vehicle-based computer 300. The vehicle-based computer 300 includes an OBDII
14 interface 312, a GPS interface 314, and a video interface 316 that interface with
15 the OBDII system 306, the GPS system 308, and the video system 310,
16 respectively.

17 The OBDII interface 312 interfaces with the OBDII system 306 via a Data
18 Link Connector (DLC), which is physical connector specified in the OBDII
19 specification. The OBDII interface 312 retrieves the raw OBDII data from the
20 OBDII system 306 in real time. The OBDII interface 312 may then format and
21 store the OBDII data in the vehicle information store 302 for presentation or use
22 with other system data. The OBDII interface 312 can also update a set of OBDII
23 error codes and events as the OBDII standard changes.

24 With regard to the GPS interface 314, location data from the GPS system
25 308 is received by the GPS interface 314 and formatted and stored for presentation

1 and/or use with other vehicle system data. The GPS interface 314 may periodically
2 store the location data in memory with a timestamp obtained from a clock in the
3 vehicle-based computer 300. The GPS interface 314 can update map information,
4 including Geographic Information System (GIS) data, which can be provided by
5 the server application 304. One particular application that can serve as the GPS
6 interface 314 is MAPPOINT by MICROSOFT Corporation.

7 The video interface 316 receives image data from the video system 310 and
8 stores the image data in the vehicle information store 302. The image data may be
9 stored with a timestamp for later retrieval and/or association with other vehicle
10 system data. The amount of image data that can be store may depend on the
11 amount of memory available in the vehicle information store 302, and is typically
12 implementation specific.

13 Other vehicle systems 318 are other vehicle systems that may generate or
14 use data during operation. For example, the other vehicle systems 318 can include
15 a vehicle security system, an obstacle detection system, media systems, vehicle
16 environment systems (e.g., temperature control), and sound systems. Other
17 interfaces 320 are provided as necessary for interfacing with other vehicle systems
18 318. Other interfaces 320 receive data from and send data to other vehicle systems
19 318. Data received from other vehicle systems 318 may be stored in the vehicle
20 information store 302.

21 One or more of the vehicle systems 306, 308, 310, and/or 318, or their
22 corresponding interfaces may be configurable. For example, a media system in the
23 other systems 318 may be configured with a list of music selections. As another
24 example, the GPS system 308 and/or the GPS interface 314 may be configured
25 with updated map, GIS, or satellite data. Such configuration data may be received

1 from a network and updated in memory, such as the vehicle information store 302.
2 The configuration data may also be downloaded from a magnetic disk, a memory
3 card, or other memory device. When configuration data is received for a particular
4 vehicle system, the appropriate interface updates the vehicle system or interface.

5 The vehicle information store 302 includes a repository for information
6 from one or more vehicle systems. One implementation of the vehicle information
7 store 302 includes a relational database. As shown, the vehicle information store
8 302 includes, but is not limited to, memory associated with each of the vehicle
9 systems shown in Fig. 3. User profiles 322 is a repository for user profile
10 information pertaining to user preferred settings. Thus, for example, user profile
11 information in the user profiles 322 may be indexable by user name or user
12 identifier. Media 324 includes media data that can be presented on a client device.
13 Exemplary media include musical tracks, other audio, and video.

14 A maintenance log 326 includes a history of vehicle maintenance. For
15 example, oil changes, repairs, and other vehicle maintenance may be recorded in
16 the maintenance log 326. Diagnostics explanations 328 are graphical and textual
17 explanations of OBDII errors and events. Because many users may not be experts
18 in car diagnostics, the graphical and textual explanations are provided to explain
19 OBDII errors and events in terms understandable by a layperson. When an OBDII
20 error or event is detected, associated graphical and/or textual explanations can be
21 retrieved from the diagnostics explanations 328 and presented to a user.

22 An OBDII data store 330 is a repository for OBDII events and errors, which
23 can be stored as errors and events are detected. The events and errors can be used
24 to identify associated diagnostics explanations 328 for presentation to a user. The
25 stored errors and events in the OBDII data store 330 can also be related to GPS

1 locations and/or map data that are stored in a GPS/map data store 332. Thus, for
2 example, a map can be presented with a marker where a particular OBDII error or
3 event was detected.

4 Video storage 334 is a repository for video images captured by the video
5 source 310. As discussed above, the video interface 316 can store captured video
6 image data in the video storage 334. Video images in the video storage 334 can be
7 presented on a display device connected to the vehicle-based computer 300 and/or
8 a display device connected to a client computer in communication with the
9 vehicle-based computer 300. Other storage 336 may be used to store any other
10 data used by the vehicle-based computer 300. For example, other storage 336 may
11 include data from other vehicle systems 318.

12 Although the vehicle information store 302 is depicted as a relational
13 database in Fig. 3, it is to be understood that any type of memory configuration can
14 be used to implement the vehicle information store 302. By way of example, and
15 not limitation, the vehicle information store 302 can be implemented using a solid
16 state memory, flash memory, and memory cards.

17 The server 304 provides services and interfaces to a client 338 for accessing
18 and/or updating vehicle information storage 302. The server 304 communicates
19 with the client 338 via a network communication port. The client 338 may be
20 either remote or local. Exemplary local and remote clients 338 are described
21 above with respect to Fig. 1 and Fig. 2.

22 The server 304 provides data according to the network protocol such that
23 data from the vehicle can be distributed to the client 338 over the network. The
24 server 304 presents a user interface 340 through which a user at the client 338 can
25 interface with the server 304. One implementation of the user interface 340 is a

1 network document 342, such as a web page, that is browsable by a browser
2 application executing on the client 338. A network document includes text and/or
3 other data organized according to a markup language that is readable by a network
4 document reader, such as a browser. Popular network document markup languages
5 are Hypertext Markup Language (HTML), Standard Generalized Markup
6 Language (SGML), and Extensible Markup Language (XML).

7 The user interface 340 can include selectable symbols, such as hyperlinks to
8 other web pages 342, which are also browsable by the client 338. In addition to
9 hyperlinks, the user interface 340 and other web pages 342 can include other
10 selectable and non-selectable symbols, such as images, graphics, text, text entry
11 fields, and tables.

12 The other web pages 344 can include information from the vehicle
13 information storage 302. In one implementation, the web server 304 includes a
14 common gateway interface (CGI) for responding to network requests by creating a
15 web page. The web server 304 can, for example, populate an HTML template web
16 page with OBDII error and event codes, along with a time of each error and event
17 code. In another implementation, the web server 304 can use an active server
18 pages application 344 to generate the web page(s) 342.

19 One exemplary implementation of an active server pages application 344 is
20 ASP .NET produced by MICROSOFT Corporation. The web page(s) 342 can
21 include embedded objects, such as Flash video clips and web controls from a
22 runtime engine. The .NET runtime engine from MICROSOFT is one such runtime
23 engine that can interact with the server 304 to facilitate creation and distribution of
24 web pages from a vehicle-based computer.
25

1 In addition, using an internet protocol (IP) address for the server 304, the
2 client 338 can request data from the server 304. The server 304 may include
3 database server functionality, by which the server 304 can query the vehicle
4 information storage 302 to satisfy client 338 requests. The server 304 includes
5 relational functionality whereby one type of data from the vehicle information
6 storage 302 can be related to and presented with other types of data from the
7 vehicle information storage 302.

8 Fig. 4 illustrates an exemplary vehicle system data scheme whereby data
9 from two different vehicle systems in a vehicle can be related for presentation to a
10 user. As shown, an on-board diagnostics (OBD) system 402 collects diagnostics
11 data, such as events and errors and stores them in an exemplary diagnostics log
12 404. Also shown is a global positioning system (GPS) 406 that collects GPS data,
13 such as position or location data, and stores them in an exemplary location log 408.

14 The diagnostics log 404 includes a code column 410 that includes one or
15 more data fields for storing diagnostics codes related to events or errors that are
16 detected by the OBD system 402 in the vehicle. The diagnostics log 404 also
17 includes a time column 412 having data fields for storing timestamps indicating
18 when associated diagnostics codes occurred. Thus, for example, an error having
19 code P0534 was detected at 9:56. Diagnostics codes in the code column 410 are
20 typically specified by a diagnostics specification, such as the OBDII standard. The
21 diagnostics codes may be specific to the make, model, or type of vehicle. The
22 timestamps in the time column 412 can be given in any time format, such as a
23 twelve hour clock or twenty-four hour clock.

24 The location log 408 includes a location column 414 and a time column
25 416. The location column 414 has data fields for storing location information

1 gathered by the GPS 406. The time column 416 includes data fields for storing
2 timestamps indicating when the vehicle was at the locations in the location column
3 414. The location data in the location column 414 may be in any geographic data
4 format, such as minutes and seconds, or decimal. As shown in Fig. 4, the
5 exemplary location data specifies latitude and longitude in a decimal format (e.g.,
6 34.05, -118.45).

7 A vehicle data management module 420 can read the data from the
8 diagnostics log 404 and the location log 408 and create relationships between the
9 location data and the diagnostics data. For example, the vehicle data management
10 module 420 can determine the location of the vehicle when a particular vehicle
11 error occurred. As illustrated, the error code P0534 occurred at 9:56 when the
12 vehicle was located at 34.05, -118.45. The vehicle data management module 420
13 can associate a location with a code and transmit the location to a mapping
14 application. The mapping application can present a marker on a map at the
15 location to indicate where a particular diagnostics error was detected. The vehicle
16 data management module 420 can be implemented with a relational database
17 software application.

18 Although the diagnostics log 404 and the location log 408 are illustrated as
19 relational tables, it is to be understood that the actual data need not be stored or
20 manipulated in a table format. For example, in a particular implementation, an
21 Application Specific Integrated Circuit (ASIC) may be used that has inputs for
22 vehicle diagnostic codes and hardware mappings to one or more of the pieces of
23 data shown in Fig. 4. In another implementation, software data structures, such as
24 linked lists, objects, or others, can be used to create relations between vehicle
25 system data and other useful data.

1 Fig. 5 illustrates a screen shot of an exemplary web page 500 that can be
2 generated by a vehicle-based server. The web page 500 can be displayed by a web
3 browser application (e.g., INTERNET EXPLORER from MICROSOFT) on a
4 remote client. When connected to the Internet or any other standard network, the
5 server may be accessed using a unique identifier (ID), for example Internet
6 Protocol (IP) address. The connection can be made the same way as currently we
7 connect to any internet server. To illustrate how a connection may be made, a
8 remote user can type, for example, <http://www.mycarname.com> or <http://127.0.0.1>
9 in the browser. After connecting to the server over the Internet, the remote user
10 can request, upload, download, and otherwise manage in-vehicle data.

11 The web page 500 includes user-selectable symbols, such as a route
12 tracking button 502, a media player button 504, a map update button 506, a
13 diagnostic history button 508, and a profiles button 510, which enable the user to
14 manage vehicle systems. When the user selects one of the buttons, a network
15 request is sent to a vehicle-based server for information related to the selected
16 button.

17 In response to the network request, the vehicle-based server provides a web
18 page related to the selected vehicle system. As illustrated, a diagnostic history
19 table 512 provides the user with a history of diagnostic events along with the
20 locations and times of the events. For example, at 9:56 on 8th Street, the vehicle's
21 antilock breaking system (ABS) was engaged. Also provided in the web page 500
22 is a map 514 showing roads traveled on the route (darker lines).

23 24 **Exemplary Operations**

1 Fig. 6 is an operation flow 600 having exemplary operations the may be
2 performed by a vehicle-based computer for remotely managing vehicle systems in
3 a vehicle. The exemplary operations in the operation flow 600 may be performed
4 periodically while the vehicle is being operated. While the exemplary operations
5 are illustrated in a particular sequence in Fig. 6, it is to be understood that the
6 exemplary operations can be performed in other sequences other than the sequence
7 shown in Fig. 6, depending on the particular implementation.

8 Prior to the operation flow 600, it is assumed that vehicle system data has
9 been gathered from one or more vehicle systems. Gathering vehicle system data
10 involves requesting vehicle system data from the one or more vehicle systems in
11 real-time. The vehicle system data may be formatted and/or stored in a memory in
12 the vehicle-based computer where the data is accessible to subsequent operations
13 in the operation flow 600.

14 A receiving operation 602 receives a network request for at least a subset of
15 the vehicle system data. The network request may come from a remote client or a
16 local client. The request is typically is formatted according to a network protocol
17 such as a TCP/IP or HTTP protocol, and has a network identifier (e.g., and Internet
18 Protocol (IP) address) associated with the vehicle-based computer. The receiving
19 operation 602 recognizes the request as being directed to the vehicle-based
20 computer, decodes the request, and identifies which vehicle system data is being
21 requested. The receiving operation 602 is optional.

22 If a network request is received for vehicle system data and/or enhanced
23 vehicle event information, a verifying operation 604 verifies the validity of the
24 network request. In one implementation of the verifying operation 604, the
25

1 network request is decrypted. Verifying may also involve validating the identity of
2 the requesting client.

3 The retrieving operation 606 retrieves vehicle system data from the vehicle
4 system data from memory. The retrieving operation 606 may retrieve "standard"
5 vehicle system data of predetermined types. For example, the vehicle-based
6 computer may automatically retrieve all OBD codes so that the OBD codes can be
7 presented to a user. Alternatively, the retrieving operation 606 may retrieve vehicle
8 system data that was requested in the receiving operation 604.

9 The generating operation 608 generates one or more network documents,
10 such as web pages, that include subsets of the vehicle system data that were
11 gathered. The generating operation 608 may generate "standard" network
12 documents with predetermined subsets of the vehicle system data. Alternatively, or
13 in addition, the generating operation 608 may generate one or more network
14 documents with requested vehicle system data specified in a network request
15 received in the receiving operation 606.

16 One implementation of the generating operation 608 involves using a
17 common gateway interface (CGI) to dynamically generate a hypertext markup
18 language (HTML) web page having vehicle system data. The vehicle system data
19 included in the HTML web page can be a predetermined subset of the vehicle
20 system data that was gathered from the vehicle systems. Alternatively, the vehicle
21 system data included in the HTML can be selected based on a network request for
22 the data.

23 Another implementation of the generating operation 608 involves
24 generating active server pages (ASP) that include the vehicle system data. An ASP
25 application may enable more variation in the types of vehicle system data that are

1 presented in the web page, as well as more flexibility in the presentation format of
2 the vehicle system data.

3 An encrypting operation 610 encrypts the generated network document to
4 achieve some level of information security. Examples of encrypting algorithms
5 that may be employed by the encrypting operation 610 are data encryption standard
6 (DES), RSA, and hashing algorithms.

7 A providing operation 612 makes the generated network document(s)
8 available to network document reader applications, such as browsers. The
9 providing operation 612 may transmit one or more network documents over the
10 network according to the network protocol. For example, the providing operation
11 612 can transmit web pages over the Internet to a client where the web pages can
12 be viewed by a browser.

13 Fig. 7 is another operation flow 700 illustrating exemplary operations for
14 configuring a vehicle system with vehicle system data. The operation flow 700
15 may be advantageously executed to update map data, media, user configuration
16 data, vehicle settings, and other vehicle data. For example, the latest geographic
17 information system (GIS) data may be downloaded for a city that the driver visits
18 frequently. The operation flow 700 can be carried out by a vehicle-based server
19 computer or other vehicle-based computer, such as the vehicle-based computers
20 illustrated and described in Fig. 2.

21 Referring specifically to the operation flow 700, a receiving operation 702
22 receives vehicle system configuration data in a vehicle-based computer. The
23 vehicle system configuration data can be received over a network or received from
24 a device or media in the vehicle. For example, the vehicle system configuration
25 data can be received from another vehicle-based computer, a remote computer, a

1 remote server, a portable computing device, a portable memory media, or other
2 devices or media.

3 In a distributing operation 704, the received vehicle system configuration
4 data is distributed to a vehicle system associated with the received vehicle system
5 configuration data. Distributing the received data may include storing the received
6 data in a memory location used by the associated vehicle system. Thus, for
7 example, media that is received from a network can be stored in a memory area
8 used by an entertainment system. As another example, map or GIS data can be
9 stored in a memory area used by a mapping application. As yet another example,
10 diagnostics code data can be used to initialize actions in an on-board diagnostic
11 system. The distributing operation 704 may format the received configuration data
12 according to a format readable by the associated vehicle system.

13 A configuring operation 706 configures the vehicle system or interface
14 corresponding to the received configuration data. During the configuring
15 operation 706, vehicle systems and interfaces are notified that there is new
16 configuration data to consume. In response, the notified systems and interfaces
17 will read the configuration data so that the systems and interfaces are up to date.
18 The configuring operation 706 may be optional depending on vehicle system
19 implementation, protocols, and/or design. The operation 700 ends at a return
20 operation 708.

21
22 **Exemplary Computer System that may be used to Implement a Remote**
23 **Vehicle Management System**

24 Fig. 8 and the corresponding discussion are intended to provide a general
25 description of a suitable computing environment in which the described

1 arrangements and procedures for managing vehicle systems may be implemented.
2 Exemplary computing environment 820 is only one example of a suitable
3 computing environment and is not intended to suggest any limitation as to the
4 scope of use or functionality of the described subject matter. Neither should the
5 computing environment 820 be interpreted as having any dependency or
6 requirement relating to any one or combination of components illustrated in the
7 exemplary computing environment 820.

8 The exemplary arrangements and procedures to transport computer data
9 between interconnected devices are operational with numerous other general
10 purpose or special purpose computing system environments or configurations.
11 Examples of well known computing systems, environments, and/or configurations
12 that may be suitable for use with the described subject matter include, but are not
13 limited to, personal computers, server computers, thin clients, thick clients, hand-
14 held or laptop devices, multiprocessor systems, microprocessor-based systems,
15 distributed computing environments such as server farms and corporate intranets,
16 and the like, that include any of the above systems or devices.

17 The computing environment 820 includes a general-purpose computing
18 device in the form of a computer 830. The computer 830 may include and/or serve
19 as an exemplary implementation of a vehicle-based computer for remote vehicle
20 system management described above with reference to Figs. 1-7. The computer
21 830 may also include and/or serve as an exemplary implementation of a network-
22 based client, server, or other computing device that may be in communications
23 with the vehicle-based computer. The components of the computer 830 may
24 include, by are not limited to, one or more processors or processing units 832, a
25

1 system memory 834, and a bus 836 that couples various system components
2 including the system memory 834 to the processor 832.

3 The bus 836 represents one or more of any of several types of bus
4 structures, including a memory bus or memory controller, a peripheral bus, an
5 accelerated graphics port, and a processor or local bus using any of a variety of bus
6 architectures. By way of example, and not limitation, such architectures include
7 Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA)
8 bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA)
9 local bus, and Peripheral Component Interconnects (PCI) bus also known as
10 Mezzanine bus.

11 The computer 830 typically includes a variety of computer readable media.
12 Such media may be any available media that is accessible by the computer 830,
13 and it includes both volatile and non-volatile media, removable and non-removable
14 media.

15 The system memory includes computer readable media in the form of
16 volatile memory, such as random access memory (RAM) 840, and/or non-volatile
17 memory, such as read only memory (ROM) 838. A basic input/output system
18 (BIOS) 842, containing the basic routines that help to communicate information
19 between elements within the computer 830, such as during start-up, is stored in
20 ROM 838. The RAM 840 typically contains data and/or program modules that are
21 immediately accessible to and/or presently be operated on by the processor 832.

22 The computer 830 may further include other removable/non-removable,
23 volatile/non-volatile computer storage media. By way of example only, Fig. 8
24 illustrates a hard disk drive 844 for reading from and writing to a non-removable,
25 non-volatile magnetic media (not shown and typically called a "hard drive"), a

1 magnetic disk drive 846 for reading from and writing to a removable, non-volatile
2 magnetic disk 848 (e.g., a “floppy disk”), and an optical disk drive 850 for reading
3 from or writing to a removable, non-volatile optical disk 852 such as a CD-ROM,
4 DVD-ROM or other optical media. The hard disk drive 844, magnetic disk
5 drive 846, and optical disk drive 850 are each connected to bus 836 by one or more
6 interfaces 854.

7 The drives and their associated computer-readable media provide
8 nonvolatile storage of computer readable instructions, data structures, program
9 modules, and other data for the computer 830. Although the exemplary
10 environment described herein employs a hard disk, a removable magnetic disk 848
11 and a removable optical disk 852, it should be appreciated by those skilled in the
12 art that other types of computer readable media which can store data that is
13 accessible by a computer, such as magnetic cassettes, flash memory cards, digital
14 video disks, random access memories (RAMs), read only memories (ROM), and
15 the like, may also be used in the exemplary operating environment.

16 A number of program modules may be stored on the hard disk, magnetic
17 disk 848, optical disk 852, ROM 838, or RAM 840, including, by way of example,
18 and not limitation, an operating system 858, one or more application programs
19 860, other program modules 862, and program data 864. Application programs
20 860 may include a remote vehicle system management application for remotely
21 accessing, uploading, or downloading vehicle system data to and from a vehicle-
22 based computer.

23 A user may enter commands and information into the computer 830 through
24 optional input devices such as a keyboard 866 and a pointing device 868 (such as a
25 “mouse”). Other input devices (not shown) may include a microphone, joystick,

1 game pad, satellite dish, serial port, scanner, or the like. These and other input
2 devices are connected to the processing unit 832 through a user input interface 870
3 that is coupled to the bus 836, but may be connected by other interface and bus
4 structures, such as a parallel port, game port, or a universal serial bus (USB).

5 An optional monitor 872 or other type of display device is connected to the
6 bus 836 via an interface, such as a video adapter 874. In addition to the monitor,
7 personal computers typically include other peripheral output devices (not shown),
8 such as speakers and printers, which may be connected through output peripheral
9 interface 875.

10 The computer 830 may operate in a networked environment using logical
11 connections to one or more remote computers, such as a remote computer 882.
12 The remote computer 882 may include many or all of the elements and features
13 described herein relative to the computer 830. The logical connections shown in
14 Fig. 8 are a local area network (LAN) 877 and a general wide area network
15 (WAN) 879. In a vehicle-based computer, the LAN 877 and/or the WAN 879 are
16 typically composed of wireless networks. In a network-based computing device,
17 the LAN 877 and/or WAN 879 may be composed of wired networks, wireless
18 networks, or any combination of wired or wireless networks. Such networking
19 environments are commonplace in offices, enterprise-wide computer networks,
20 intranets, and the Internet.

21 When used in a LAN networking environment, the computer 830 is
22 connected to the LAN 877 via a network interface or an adapter 886. The network
23 interface 886 provides communications services for transmitting and receiving
24 data to and from one or more clients. For example, the network interface 886
25 formats, encodes, modulates, demodulates, and decrypts data communicated via

1 the LAN 877. The network interface 886 operably communicates over a network
2 using a standard network communication protocol. Examples of communications
3 devices suitable for the network interface 886 in a vehicle-based server computer
4 include a cellular modem, Wireless Fidelity (WiFi), or other wireless
5 communications devices.

6 The network adapter 886 may also be used to facilitate communications in a
7 WAN 879 networking environment. As such, the computer 830 typically
8 communicates via the network adapter 886 or other means for establishing
9 communications over the WAN 879. The network adapter 886, which may be
10 internal or external, may be connected to the system bus 836 via the user input
11 interface 870 or other appropriate mechanism. Depicted in Fig. 8 is a specific
12 implementation of a WAN via the Internet.

13 In a networked environment, program modules depicted relative to the
14 personal computer 830, or portions thereof, may be stored in a remote memory
15 storage device. By way of example, and not limitation, Fig. 8 illustrates remote
16 application programs 889 as residing on a memory device of remote computer 882.
17 It will be appreciated that the network connections shown and described are
18 exemplary and other means of establishing a communications link between the
19 computers may be used.

20 Although some exemplary methods, devices and exemplary systems have
21 been illustrated in the accompanying drawings and described in the foregoing
22 detailed description, it will be understood that the methods and systems are not
23 limited to the exemplary embodiments disclosed, but are capable of numerous
24 rearrangements, modifications and substitutions without departing from the spirit
25 set forth and defined by the following claims.